LISP Pointers

VOLUME 2, NUMBER 1
JULY-AUGUST-SEPTEMBER
1988

TABLE OF CONTENTS

EDITORIAL BOARD

This issue of the Lisp Pointers newsletter has been funded by I.N.R.I.A. and ACM SIGPLAN. Neither group has directed or controlled its publication. The opinions expressed herein are solely those of the authors, editors, publisher and other contributors and do not necessarily reflect the opinions of I.N.R.I.A or ACM SIGPLAN or the opinions of companies affiliated with individuals involved with this effort.

Dear Colleague,

The time of year seems right to offer Thanksgiving that you're finally getting this copy of Lisp Pointers. We hope the next one will be out in a more timely fashion. The Dribble File contains two reviews of video tapes. If anyone knows of any other video tapes out there, please let me know. You'll find a new column, "Readings in Scheme" by Ozan Yigit. I'm especially pleased with this since we've always meant this to be both a Lisp and Scheme newsletter and it's good to see more emphasis on Scheme. As always, department heads are dependent upon contributions from you so, please, think what you might have to share with all of us. Jokes, anecdotes, or cartoons, anyone?

We are presently looking into the possibility of having Lisp Pointers taken in as a special publication (like FORTRAN Forum) under ACM SIGPLAN. The major advantage is that we would have an assured level of funding for future newsletters. Sponsors have not been as easy to find as we had hoped. It's been a hard year for many companies in the Lisp and Scheme community. To go under the ACM umbrella, there are three potential routes: becoming a separate special interest group (like SIGFORTH), becoming a special publication under SIGPLAN (like FORTRAN Forum); or starting as a special publication with the intention of eventually separating as a SIG (like STAPL --> SIGAPL and AdaTEC --> SIGAda).

When I was editor of Ada LETTERS, we usually had a backlog of material which we would hold for succeeding issues. That publication started quarterly and moved to bimonthly to handle the submissions. SIGPLAN Notices also has a tradition of having a good-sized backlog. Lisp Pointers has never been in that position. Our department chairs tend to search out material and we hold an issue until we have enough material to fill it. It seems that members of our community tend more toward programming than writing (hence, the new Algorithms section). What this is leading to is the thought that there might not be enough organizational interest out there to support a separate organization and that continuing only as a newsletter might be the wiser move.

As always, the publication is yours so I would welcome comments, questions, and alternative suggestions.

Our thanks to ACM SIGPLAN and I.N.R.I.A. for sponsoring this issue. There is a call for sponsors for future issues. I.N.R.I.A. has been kind enough to sponsor all non-US mailings. This means that we only need sponsorship for the US mailings. The bottom line for those who don't want to read the page of text is that you can work out the costs of sponsorship for US subscribers by looking at what it costs your duplicating department to make 1300 copies of this issue and what it costs your mail room to mail 1100 copies of approximately 60 pages. Contact me if you have any additional questions.

Sincerely,

Mary S. Van Deusen

Mary S. Van Deusen, Editor

11/7/88

Aug. 10, 1988

Dear Ms. Van Deusen,

I just received the April-May-June 1988 edition of LISP Pointers. When I started to read Pavel Curtis' article on algorithms, it brought to mind my dilemma.

How are LISP algorithms specified ? In particular, how are function specifications written (flowcharts, Jackson methods, etc) ?

I have not written any function specifications. I have only made the I/Fs & what the function does clear (ala CLtL).

My dilemma ? My medieval managers & archaic customers have been raving about the lack of documentation. They are, of course, used to filing cabinets full of function specifications complete with flowcharts, etc.

I wrote to you because there doesn't seem to be any appropriate department. If you have any suggestions, please let me know.

Thank you.

Yours truly,

Harlen Seo

c/o SRA
Hirakawa-cho 1-1-1
Chiyoda-ku
Tokyo 102

20-B Gilstead Road
Singapore 1130

Mary S. Van Deusen, Editor
IBM Research
P.O. Box 704
Yorktown Heights, NY 10598

LISP NEWSLETTER

Dear Ms. Van Deusen,

Last year I attended the Advanced Common LISP Programming tutorial given at AAAI-87 by David Touretzky and Richard Gabriel. The tutorial manual mentioned a newsletter concerning LISP of which you are the Editor.

I am very interested in LISP, and am currently engaged in an MsC research project at the National University of Singapore of which the thesis topic is **Design and Optimization of LISP Interpreters**. (At age 60, I believe I am their oldest student!) Also I have also just finished a 500+ page book entitled **LISP Programming for Artificial Intelligence: A Common LISP Tutorial**, which Addison-Wesley will be publishing in early 1989. This will hopefully be followed a year or so later with a second volume on LISP applications projects.

I would appreciate it if you would be so kind as to include my name on the mailing list for the newsletter.

Yours sincerely,

David J. Steele

(No relation to Guy L. Steele, Jr!)

For ordering information for David Steele's book please contact:

Addison-Wesley
Route 128
Reading, MA 01867
(617) 944-2500

DELPHI CLOS Activity

DELPHI announced the inclusion of the Common Lisp Object System in their implementation of Common Lisp. They describe three uses of this system: a kernel for KRS, the Knowledge Representation System developed at the University of Brussels; a Persistent Object System that interfaces to a variety of underlying database ssytems, and an "atomic" metaclass which aids concurrent programming. For further information, contact:

Delphi S.p.A.
V.le Milanofiori, Pal. F1
I-20090 Assago (MI)
ITALY

AI Conference

The International Computer Science Conference '88 will be held in Hong Kong, December 19-21, 1988. The topic will be "Artificial Intelligence: Theory and Applications." The conference is sponsored by the Computer Society of the IEEE, Hong Kong Chapter. Paper submissions are due June 15, 1988. For submission information, contact:

Jean-Louis Lassez
IBM Watson Research
PO Box 704 (H1-A12)
Yorktown Heights, NY 10598

For conference information, contact:

Computer Society of IEEE, Hong Kong Chapter
G.P.O. Box 5318
Hong Kong

Addison-Wesley Video Courses

Addison-Wesley has produced a series of video courses on Artificial Intelligence. I received 40 minutes out of two hours of video for the course, "Artificial Intelligence: Foundations and Applications", narrated by Patrick Henry Winston. The 3/4" videos consist of three modules: Perspectives on Artificial Intelligence, Representation and Expert Systems, and The Future of AI Technology. Along with the video tapes, the student is sent workbooks to use to reinforce the material being described. The assumption is that these videos will be used for in-house courses taught by a corporate instructor. For that reason, an Instructional Guide is also included.

There seem to be two main trends in educational videos today. They are either videotaped live using multiple cameras and an in-camera switching unit to vary the views or they are created like a jigsaw puzzle of multiple pieces and pasted together in post-production editing. There are advantages and disadvantages to each approach. The videos produced by Addison-Wesley are of the post-production variety, that is, they use a main teacher (Patrick Winston) and intercut a wide variety of graphics to demonstrate what he is saying and clips of other speakers to maintain visual interest. The years of TV watching have made us a more sophisticated audience. In 1950, long scenes shot from a single angle were taken for granted. Today, we require more varied, and shorter, cutting of scenes to make us keep watching.

AI: Foundations and Applications is a thoroughly professional piece of work. The simple animation graphics are of very fine quality, the video is fast-paced, and Winston, himself, is an excellent speaker. The script was the result of long preparation, and the result is a well-structured course that presents a great deal of material in a very short period of time.

Many long years ago, when I was learning to write animated movie scripts, I was taught that people couldn't sit still for more than about 15 minutes of concentrated video teaching. From my own itchy experience, I'd still agree with that figure. The module which I received, "Representation and Expert Systems" was broken into three pieces with checkpoints which encouraged you to switch to your workbook (or go out for a drink). The combination of entertaining and educational video, as well as the breaks that eliminated guilt at increasing itchiness, made for one of the most enjoyable video learning experiences I can remember.

I must admit that I didn't use the workbook, so I can make no comments on its usability. The end of the book did have a short list of topics and recommended readings, as well as answers to the workbook questions.

The Instructional Guide lists sources of free or nominal-cost demonstration software which an instructor might want to use in conjunction with this video. It also listed several organizations for information on possibilities for academic course credit. An outline is provided which breaks the course down into a two or three day course divided between video and workbook exercises.

As I said earlier, each approach to video production has its own advantages and disadvantages. The clear disadvantage to this course is its cost, $3000. The assumption is that a company would buy the course and amortize its cost over several classes of several students. The price is based on high production costs to do this type of video.

For further information, contact:
Christa Starling, Product Manager
Addison-Wesley Training Systems
Route 128
Reading, MA 01867
(617) 944-3700 X2714

Morgan Kaufmann Video Courses

Morgan Kaufmann has come out with a set of sixteen video lectures collected as "The Stanford Computer Science Video Journal." These videos are produced in a lecture hall equipped with many cameras, an in-camera switcher, and a very experienced operator who has to second-guess where the speaker is going to go next in order to choose the best camera view.

In-camera editing has many advantages when it comes to production time and costs. Most of the cost is incurred in the setup of studio equipment. After that, the time to create a production is basically the time it takes to give the lecture plus a small amount of post-production time for titling, etc. Because of the lower production costs, the cost to you is equivalently lower -- in this case, $49.95 per VHS tape (as a Beta tape user, allow me to file an objection). This puts each tape almost in the stocking stuffer category.

Each tape is approximately an hour long. The only camera shots possible are of the speaker in closeup, the speaker in front of a blackboard, the blackboard, and the prepared presentation visuals. When you have a good natural teacher, a one hour lecture can be a joy. The first video which I have seen, "Computerized Understanding of Language" by Terry Winograd, is one such lecture. Winograd is a such a dynamic speaker that the camera operator is sometimes hard-pressed to follow him around the platform. The visuals must often have been unreadable for the audience but, on tape, they are easily frozen. There were three things I would say I really enjoyed about this video. First, Winograd has a wonderful way of integrating past, present and future research trends in the field of language understanding. Secondly, his examples were both educational and entertaining. I found I wanted to go out and repeat to people his story about green canaries. And third, it reminded me of the humor which is so particular to people involved with computers, that is, the reliance of so many of our jokes on literal interpretations of language.

For further information on this and other videos, contact:

Morgan Kaufmann Publishers
PO Box 50490
Palo Alto, CA 94303-9953

Lisp Puzzles

Christian Queinnec Jérôme Chailloux

April 16, 1988

1 First Puzzle

Jérôme and I asked our students at École Polytechnique to program the following function

```
(xpl '(a b c d))
→   ((a) (a b) (a b c) (a b c d))
```

Given a list, xpl returns the list of all its initial segments ordered by increasing length.

READERS' SOLUTIONS TO PUZZLE NUMBER 1

```
Received: from  hplabs.HP.COM by IBM.COM on 06/20/88 at 16:32:02 PDT
Received: from hplms2.HP.COM (hplms2) by hplabs.HP.COM with SMTP ; Mon, 20 Jun 88 15:30:47 PST
Received: from hplddc.HPL.HP.COM (hplddc.hpl.hp.com) by hplms2.HP.COM; Mon, 20 Jun 88 16:30:30
Received: by hplddc.HPL.HP.COM; Mon, 20 Jun 88 16:30:11 pdt
From: Dennis de Champeaux <ddc%hplddc@hplabs.HP.COM>
Message-Id: <8806202330.AA02418@hplddc.HPL.HP.COM>
Date: Mon, 20 Jun 88  16:30:04 PDT
Subject: Puzzle 1 in LISP Pointers vol 1 no 6
To: maida@ibm.com
Cc: ddc%hplddc@hplabs.HP.COM
X-Mailer: NMail [$Revision: 2.7 $]
```

I do not like the seven solutions that the authors provided.

Here is a one liner:

```
(defun xpl (l) (reverse (maplist 'reverse (reverse l) )))
```

Sincerely,

Dennis de Champeaux

To the editor:

Here is a Common Lisp solution to Queinnec and Chailloux's puzzle, LP I-6.2. Perhaps you could publish it with other reader-submitted solutions.

```
(defun xpl (x)
  (reverse (maplist #'reverse (reverse x))))
```

Larry Stabile
BBN Advanced Computers, Inc.

Solutions to the First Puzzle,   Second Puzzle

Hereafter are the seven solutions we got. They are all written in Le-Lisp[1]. The problem is now "Which one is the more efficient" ? The solution is elsewhere !

2.1 Solutions 1 and 2

The trick is to notice that, given (a b c) it is straightforward to obtain ((c) (b c) (a b c)) by a recursive function such as

```
(defun foo (l r)
   (if (consp l)
       (foo (cdr l) (cons l r))
       r ) )
```

With some reverse, one can obtain

```
(defun xpl1 (l)
   (xpl11 (reverse l) nil))
(defun xpl11 (l r)
   (if (consp l)
       (xpl11 (cdr l) (cons (reverse l) r))
       r ) )
```

xpl11 is tail-recursive, the following is not

```
(defun xpl2 (l)
   (reverse (xpl21 (reverse l))))
(defun xpl21 (l)
   (if (consp l)
       (cons (reverse l) (xpl21 (cdr l)))
       nil ) )
```

2.2 Solutions 3 and 4

The trick is to use rdc (the mirror of cdr) which, given a list, returns the list except its last term. rdc is better explained as

```
(defun rdc (l)
   (reverse (cdr (reverse l))) )
```

With rdc, the result is easily computed

```
(defun xpl3 (l)
   (xpl31 l ()))
(defun xpl31 (l r)
   (if (consp l)
       (xpl31 (reverse (cdr (reverse l))) (cons l r))
       r))
```

The following is the non-tail-recursive equivalent

```
(defun xpl4 (l)
   (reverse (xpl41 l)) )
(defun xpl41 (l)
   (if (consp l)
       (cons l (xpl41 (reverse (cdr (reverse l)))))
       nil ) )
```

[1] Le-Lisp is a trademark of INRIA.

2.3 Solutions 5 and 6

Iterators such as mapcar can be used

```
(defun xpl5 (l)
   (if (consp l)
       (cons (cons (car l) nil)
             (mapcar (lambda (r) (cons (car l) r))
                     (xpl5 (cdr l)) ) )
       nil ) )
```

But one can also rediscover mapcar and obtain

```
(defun xpl6 (l)
   (if (consp l)
       (cons (cons (car l) nil)
             (conslist (car l) (xpl6 (cdr l))) )
       nil ) )
(defun conslist (a l)
   (if (consp l)
       (cons (cons a (car l))
             (conslist a (cdr l)) ) ) )
```

2.4 Last Solution

The last one bears some resemblance to Ashcroft's definition of reverse, i.e.

```
(defun reverse (l)
   (if (consp l)
       (if (consp (cdr l))
           (cons (car (reverse (cdr l)))
                 (reverse (cons (car l)
                                (reverse (cdr (reverse (cdr l)))) )) )
           l )
       nil ) )
```

xpl7 uses four embedded calls to reverse but it works !

```
(defun xpl7 (l)
   (if (consp l)
       (reverse (cons l (reverse (xpl7 (reverse (cdr (reverse l)))))))
       nil ) )
```

Lisp Puzzles

Christian Queinnec Jérôme Chailloux

April 16, 1988

Solution to the Second Puzzle

Here are the numbers of calls to car, cdr, cons and consp. They were counted while xpl'ing a list of 100 terms. The cost of reverse and mapcar (with the following definitions) were, of course, taken into account

```
(defun reverse (l)
   (labels ((reverse (l r)
               (if (consp l)
                   (reverse (cdr l) (cons (car l) r))
                   r ) ))
         (reverse l nil) ) )
(defun mapcar (fn ll)
   (if (consp ll)
       (cons (funcall fn (car ll))
              (mapcar fn (cdr ll)) )
       nil ) )
```

The results are

	car	cdr	cons	consp
xpl1	5150	5250	5250	5352
xpl2	5250	5350	5350	5453
xpl3	10000	10100	10100	10301
xpl4	10100	10200	10200	10402
xpl5	10000	5050	10100	5151
xpl6	5150	5050	10100	5151
xpl7	20000	20100	20100	20501

The cost is minimal with the first solution.

LISP AND SYMBOLIC COMPUTATION: An International Journal

NOW AVAILABLE

Scope:

- Programming language notations for symbolic computing (e.g., data abstraction, parallelism, lazy evaluation, infinite data objects, self-reference, message-passing, generic functions, inheritance, encapsulation, protection, metaobjects).
- Implementations and techniques (e.g., specialized architectures, compiler design, combinatory models, garbage collection, storage management, performance analysis, smalltalks, flavors, common loops, etc.).
- Programming logics (e.g., semantics and reasoning about programs, types and type inference).
- Programming environments and tools (e.g., knowledge-based programming tools, program transformations, specifications, debugging tools).
- Applications and experience with symbolic computing (e.g., real-time programming, artificial intelligence tools, experience with LISP, object-oriented programming, window systems, user interfaces, operating systems, parallel/distributed computing).

Editors-in-Chief:
Richard P. Gabriel, Lucid, Inc. and Guy L. Steele Jr., Thinking Machines, Inc.

Articles of Volume 1, Issue 1 include:
Expansion-Passing Style: A General Macro Mechanism, R. Kent Dybvig, Daniel P. Friedman, Christopher T. Hayes; *OAKLISP: An Object-Oriented Dialect of Scheme*, Kevin J. Lang, Barak A. Pearlmutter; *The Mystery of the Tower Revealed: A Nonreflective Description of the Reflective Tower*, Mitchell Wand, Daniel P. Friedman; *Technical Issues of Separation in Function Cells and Value Cells*, Richard P. Gabriel, Kent M. Pitman

Submissions and more information contact:

Jan Zubkoff
Associate Editor, LASC
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
jls@lucid.com
415/329-8400

For subscriptions please contact:

Kluwer Academic Publishers
P.O. Box 358
Accord Station
Hingham, MA 02018-9990
617/871-6600

SPONSORSHIP OF LISP POINTERS

Lisp Pointers is a non-profit publication created by the Lisp community for the Lisp community. Currently, Lisp Pointers is not affilated with any organization. For this reason, it is dependent upon the sponsorship of companies interested in Lisp for its publication.

The following organizations have in the past, or are currently, sponsoring Lisp Pointers:

IBM Thomas J. Watson Research
Institut National De Rechere En Informatique Et En Automatique (I.N.R.I.A.)
Xerox Parc
Microelectronics and Computer Technology Corporation (MCC)
Texas Instruments
Digital Equipment Corporation
Mitre Corporation
ACM SIGPLAN

Lisp Pointers is a newsletter, that is, it contains technical articles which are not refereed and which, therefore, may be republished in other technically refereed journals later. Lisp Pointers is a forum for preliminary papers, as well as for the fast interchange of ideas. As well as technical articles, Lisp Pointers contains columns and departments, such as the following:
Query IO - The Scheme of Things - International News - Programming Environments
Book Reviews - Lisp Implementations - ((lambda (discussions) (report on X3J13)))

Sponsors are permanently listed on the back cover of Lisp Pointers. We do this to thank those companies who have joined us in producing a publication which we think is both needed and wanted by this important research and production community.

Because no organization is involved, the board running Lisp Pointers tends to be very conservative both legally and financially. A disclaimer for a sponsor company appears on the inside front cover. The disclaimer for the first issue reads as follows:

> This issue of the Lisp Pointers newsletter has been funded by the IBM Corporation. The IBM Corporation has not directed or controlled its publication. The opinions expressed herein are solely those of the authors, editors, publisher and other contributors and do not necessarily reflect the opinions of the sponsoring company, the IBM Corporation, or the opinions of companies affiliated with individuals involved with this effort.

Sponsorship involves the commitment to cover cost and work involved in the printing and distribution of a single issue of Lisp Pointers. The editor creates a camera-ready copy which is sent to the sponsoring company. The sponsor arranges for the duplication and mailing of the issue. Cost to the sponsoring company is dependent upon the size of the mailing list, which will increase over time, the access to inhouse duplication facilities, the size of the issue, and the cost of mailing to subscribers within the US. **INRIA has accepted permanent sponsorship of all non-US subscribers.** The number of US subscribers has grown from 500 to approximately 1100 over seven issues and it can be assumed that that growth will continue for at least the next year. In addition to the number printed and mailed, the sponsor is asked to print an additional 150-200 copies which are used to satisfy requests for backorders. Previous issues have run between 40 and 60 pages. Your mailroom should be able to estimate the cost of mailing out each issue.

At the time this letter is being sent to you, we are looking for sponsors for issues 8 and beyond. Lisp Pointers is published four times per year.

Implementation Summaries

Walter van Roggen
Digital Equipment Corporation
290 Donald Lynch Blvd. Marlboro MA 01752
vanroggen@hudson.dec.com

This list represents the descriptions I have received between 20 April 1988 and 19 August 1988.

Name Allegro Common Lisp

Standard Common Lisp

Additional Features .

Object Lisp
Macintosh windows, menus, dialogs as objects
EMACS-style editor
native-code compiler
window-based stepper, inspector, trace
can produce stand-alone applications which can be distributed without licensing fees
color QuickDraw
Flavors
Foreign Function Interface to MPW C, Pascal, and assembly language
Standalone Application Generator

Version V1.1, April 1988, $600. Allegro Flavors $350. Allegro Foreign Function Interface $200. Allegro Standalone Application Generator $600. Educational and volume discounts are available.

Hardware Apple Macintosh with at least 1 MB RAM

Support Fully supported by Coral Software

Contact Coral Software Corp, PO Box 307, Cambridge MA 02142, (800)521-1027 or (617)868-7110

Name Swifte-LISP

Standard Common Lisp subset

Additional Features .

mini-prolog interpreter
Basic Flavors
compatibility packages for UCI/Stanford Lisp, Standard Lisp and Scheme
profiler
pretty printer

Hardware M68000 machines running the MIRAGE operating system: HP200; HP300/350; Gemini Challenger; U-Micro series 1000 + 2000; Stride 400 series; Ferroglen Lucid; MicroAPL Spectrum and Aurora; Newtons Fastnet; Pinnacle 1A; Motorola, Tadpole and IMP 68000-based VME cards; and Atari ST.

Contact Sahara Software, South Bank Technopark, 90 London Road, London SE1 6LN. Telephone: 01-922-8850. Telex: 896885 IOTA